

# Refactoring Team Design

Designing teams and busting myths at REA

# Topics for today

- What does a typical REA team look like?
- Refactoring mobile teams
- Models for team collaboration
- What does the data tell us about our teams?





Wednesday, January 18, 1967

6p

TODAY'S TV PAGES 12 and 13

***THE FAB FOUR***  

---

***NOW THREE***

---

**PAUL IS DEAD**





**GOD**  
**IS DEAD**  
- Nietzsche



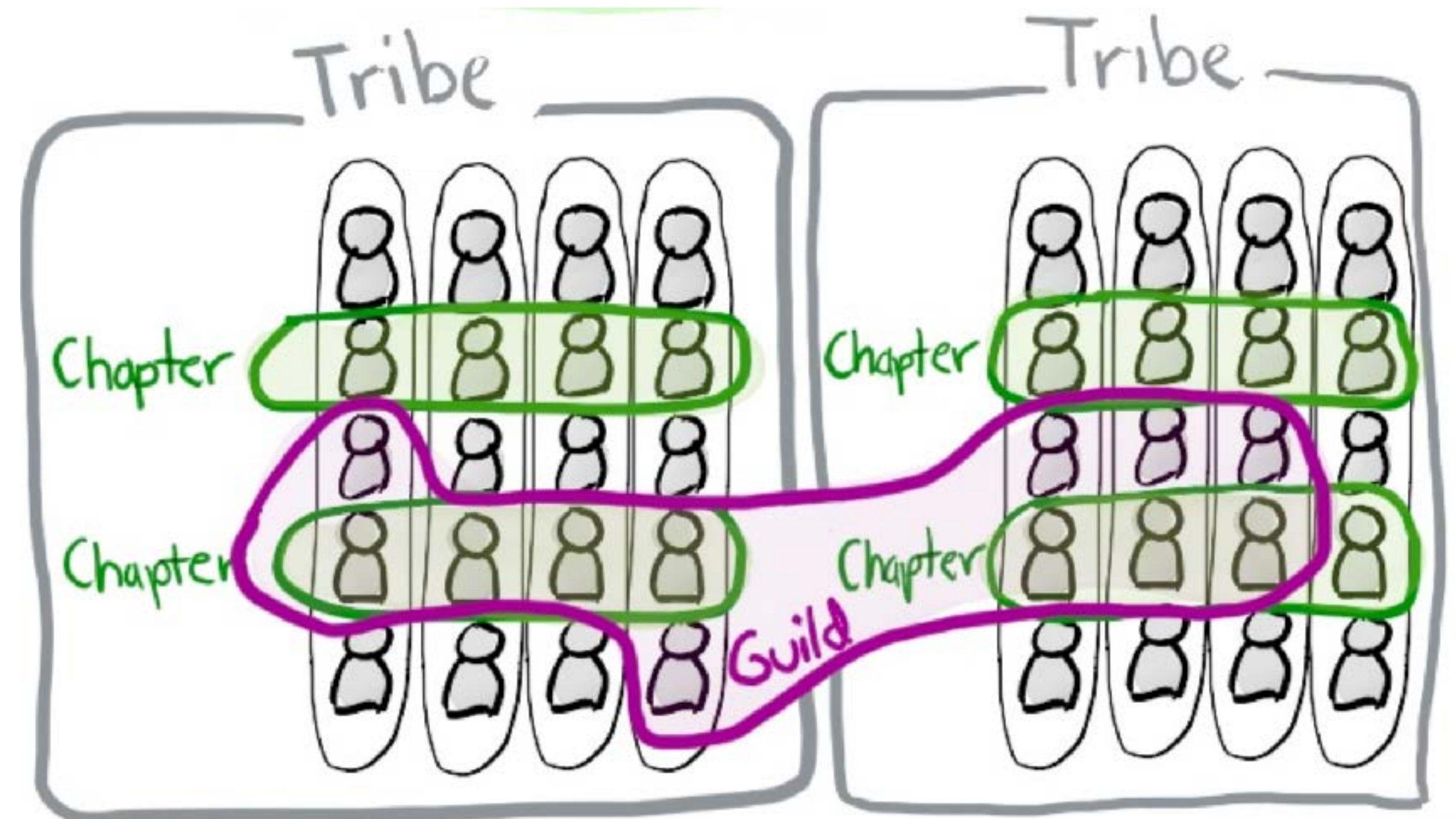
DE  
LA  
SOUL  
is dead





# Agile squads are dead

- Tribes and squads != agile
- Teams are complex and evolutionary



# Team design by imitation

“ We’ve done  
away with BAs ”

Company A

“ We don’t have  
QAs any more ”

Company B

“ We’re going  
DevOps ”

Company C

“ We use Agile  
Coaches ”

Company Z

“ Let’s copy that  
team on Level 3 ”

Us

“ We’re SAFe ”

Anonymous

# Team design by context

Key concepts to consider

- Product lifecycle  
Experimenting. Sustaining. Sunsetting.
- Distributed teams  
What is optimal for remote teams?
- Custodianship  
You build it, you run it
- People leadership  
Developing careers
- Eco-system complexity  
Integration points and dependencies
- Products and platforms  
Different mindset?
- Technology  
Optimising for skills and capabilities



# Designing teams - a complex problem



“ Stop treating teams and organisations as machines.

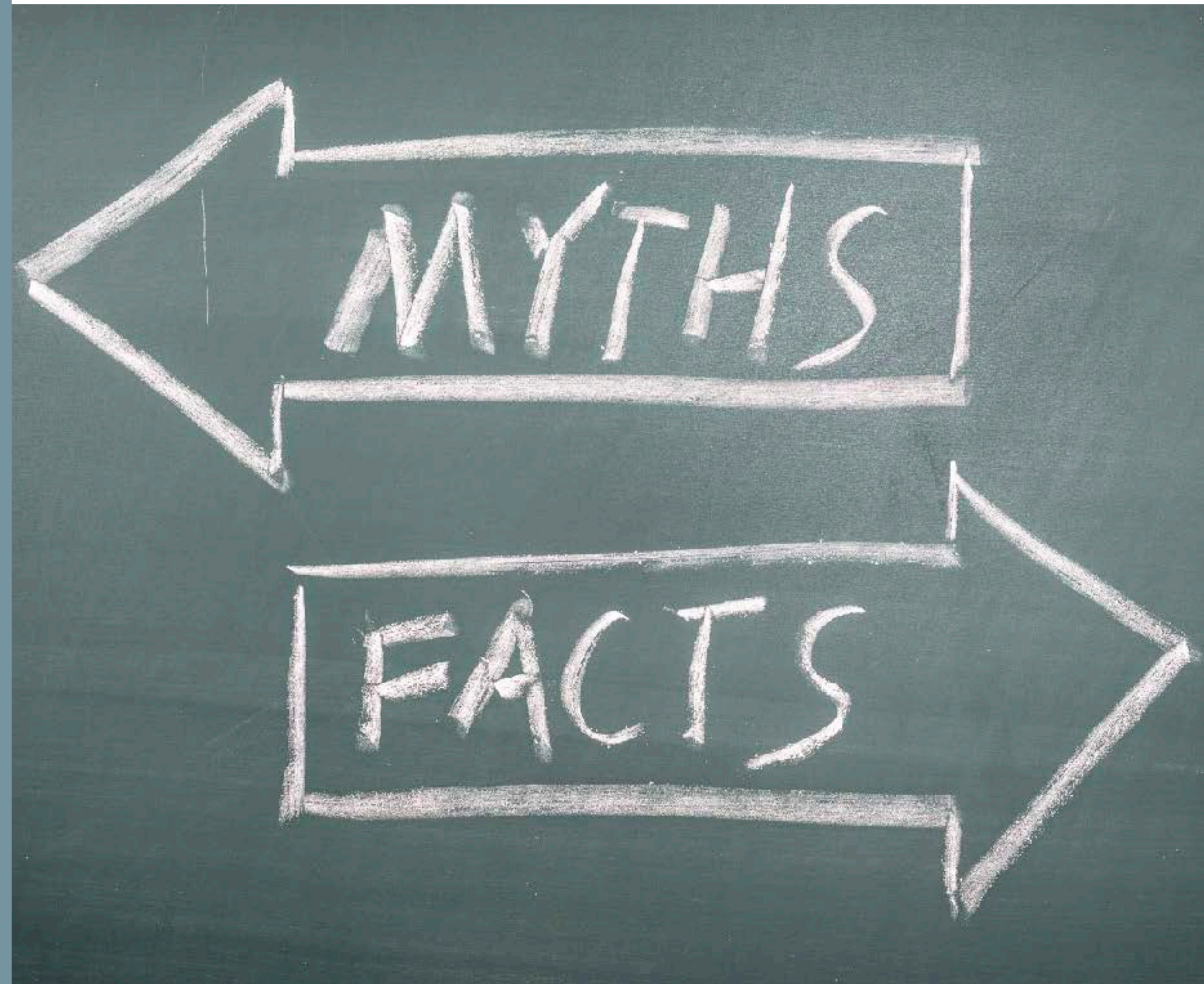
Start treating them as complex (living) systems.

”

Jurgen Appelo

# Project Dynamite: blowing up team myths at REA

- There is a canonical REA team profile (myth?)
- We have a standard model for designing new teams (myth?)
- All teams have 12 people (myth?)





# Product delivery team roles



Product Manager



QA



Designer



Developer



Delivery Lead



BA



Systems Engineer

# Product delivery tribe roles



Tech Lead



Engineering Manager



## LEADS

← SMP →

← DM →

← PM →

PROMOTE

PM

TTL

DEVS\*6

ENGAGE

PM

TTL

DEVS\*6

QA

CONNECT

PM

TTL

DEVS\*6

Scientific research

**The typical REA team?**

**There isn't one**



# The data

**44** Teams in REA Australia across 7 LoBs and 15 tribes

**13** Teams distributed with Xi'an

**11.9** Average size of a Xi'an teams

**10.3** Average size of a product delivery teams

**6.6** Average number of developers per teams

**7.6** Average number of developers in Xi'an teams

**35%** Teams with a QA

**7:1** Average Developer to QA ratio

**45%** Teams with a BA

**25%** Teams with a Systems Engineer

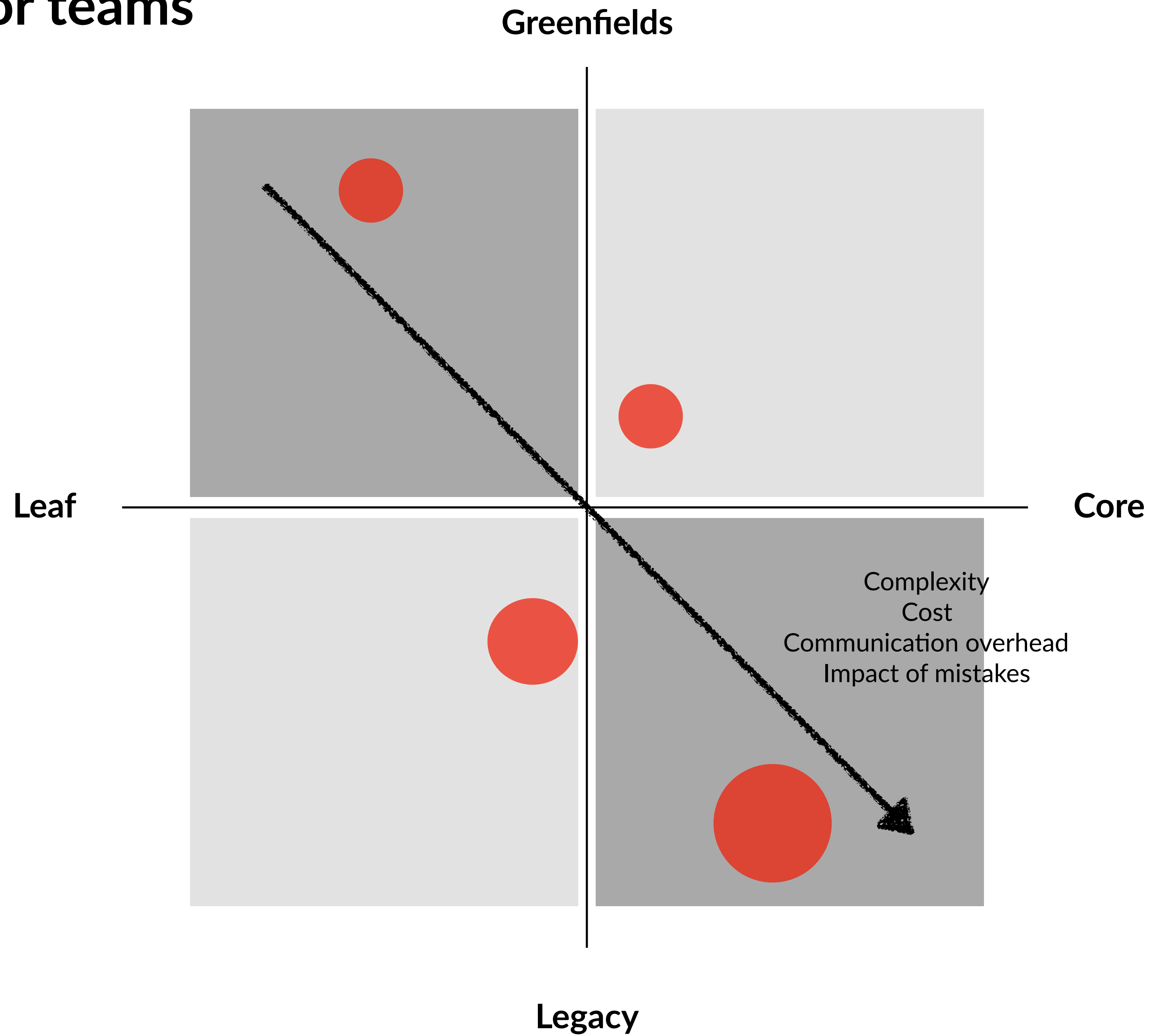


**5** Teams matching the most common profile: Delivery Lead - Designer - Developer - Product Manager

**15%** People in "tribe roles"

**5** Most common tribe roles:  
Systems Engineer  
Designer  
Technical Lead  
Engineer Manager  
Product Manager

# The context for teams





# Design teams based on your context

- Teams are complex
- There are many variables
- Determine what are you trying to optimise for



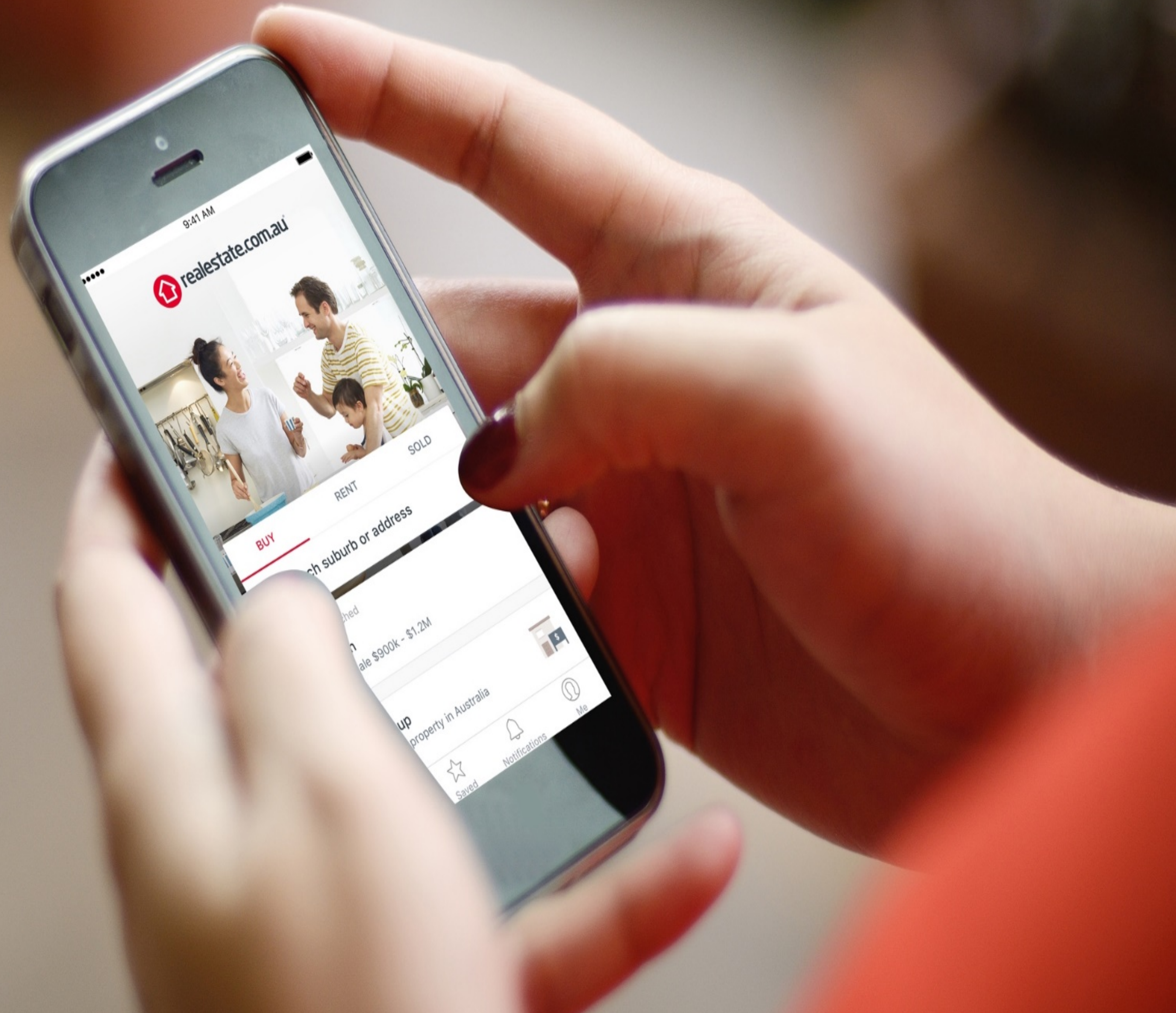
# Evolutionary teams

- We refactor code design
- We refactor system design
- We should refactor team design



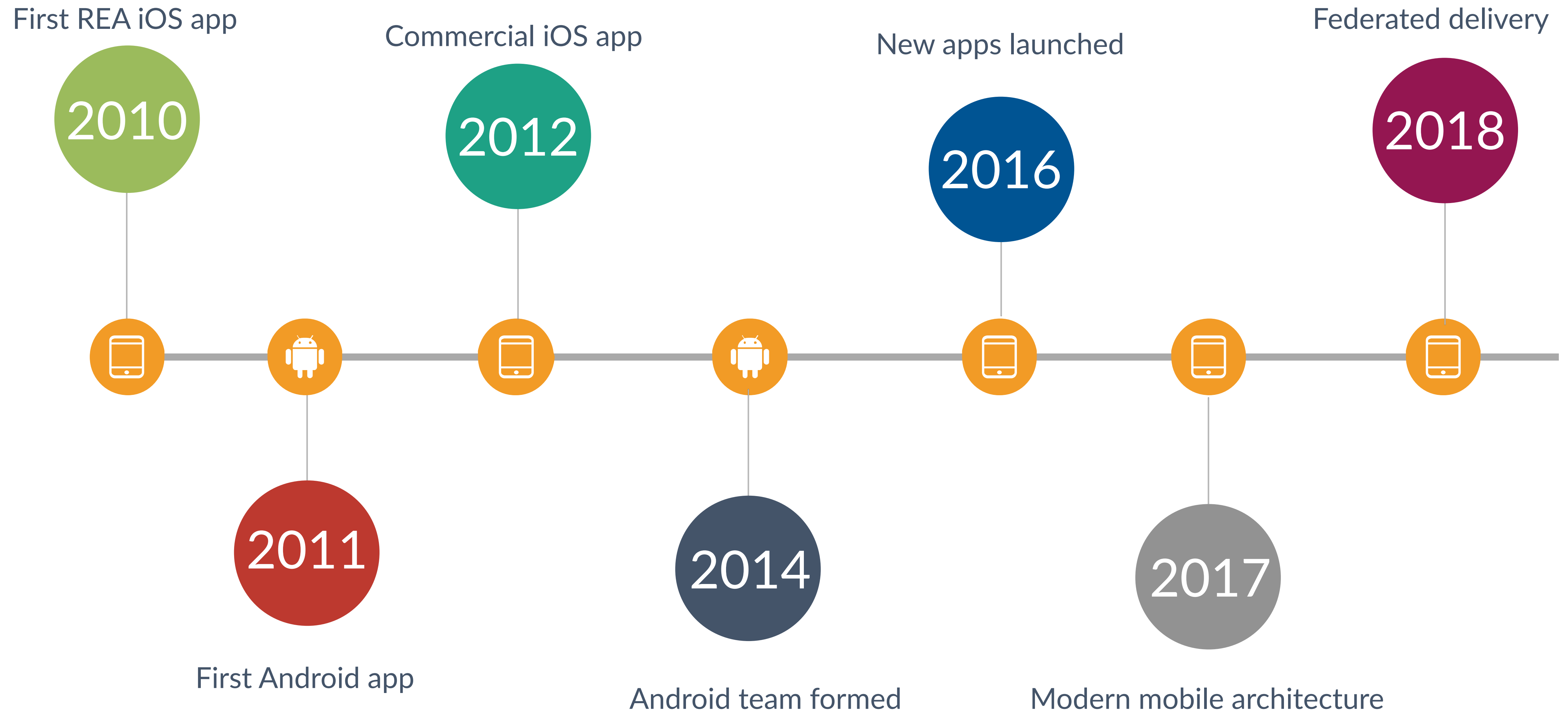


# Refactoring our mobile teams





# Mobile development @ REA



# Conway's Law



“ Organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations ”

Melvin Conway



# The unintended side effects of Conway's Law

- Our iOS and Android teams built iOS and Android apps that lead the market
- and had different features
- and looked different
- and shared little
- We built two new monoliths





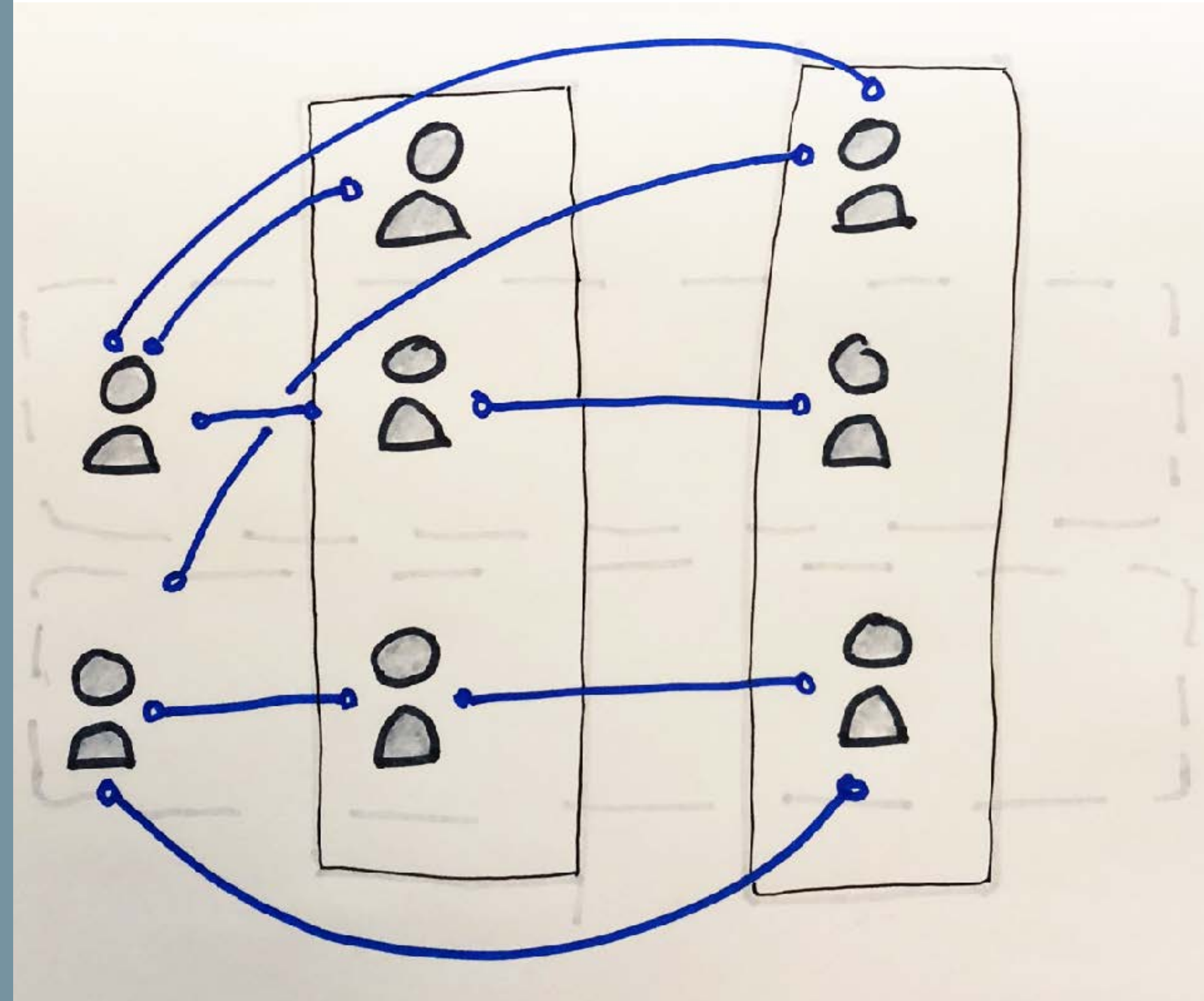
# Optimisation problems

---

- Co-ordinating a feature across platforms/teams
- No back end development skills: dependencies
- Designers: platform-focused or experience-focused?

# Experiments in refactoring team design

- Platform teams
- Feature teams
- Virtual teams



# Team design principles

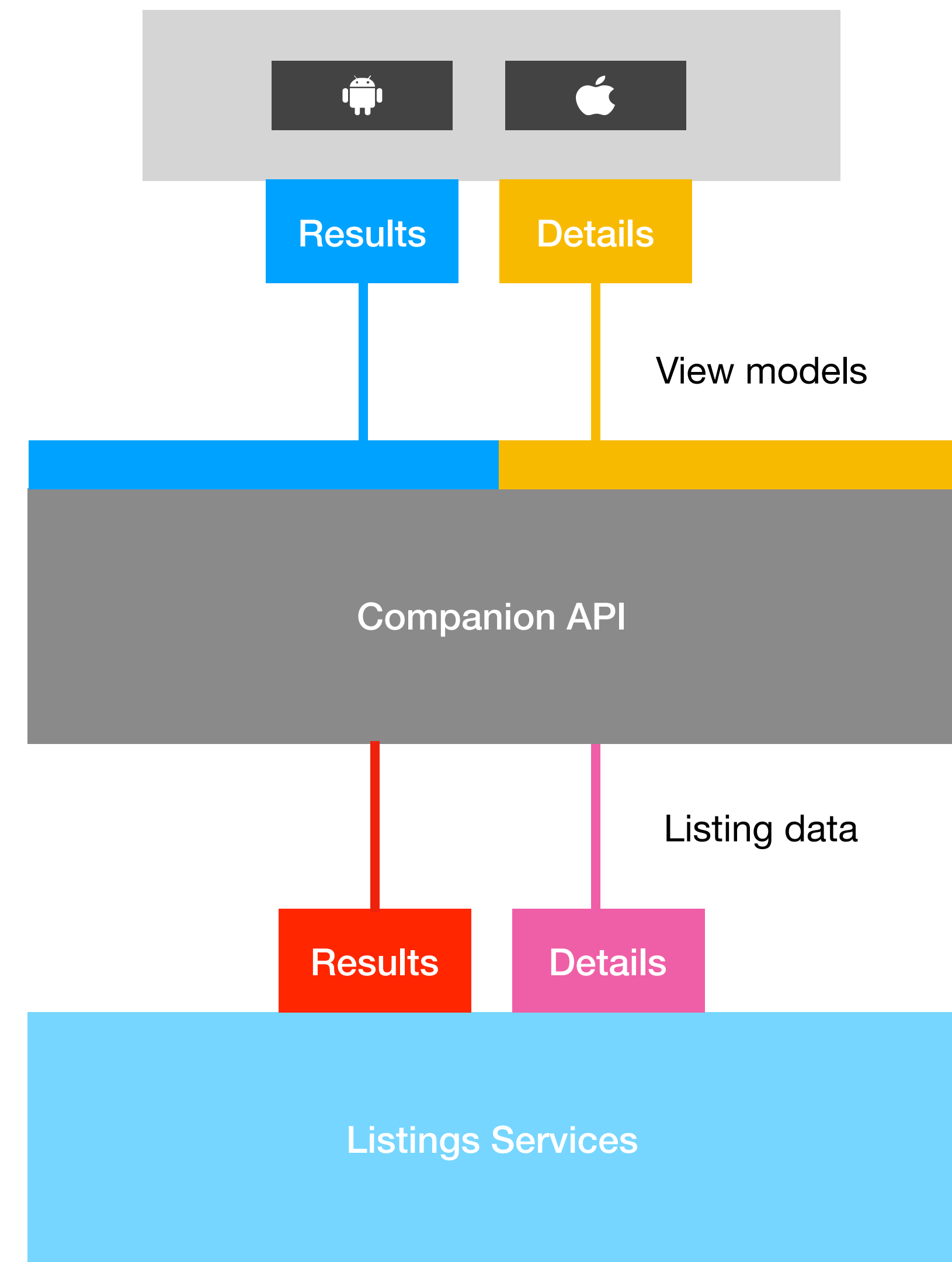
---

- Clear purpose
- Minimise dependencies
- Maximise autonomy
- Reduce communication overhead
- Remove technology silos



# Objective-driven teams

- Kaizen
- Mobile architecture



# What has architecture got to do with team design?

---

- Inverse Conway Manoeuvre



“ A few months ago I called myself an iOS developer. Now I call myself a full stack mobile dev. ”

Yanzi Li



# Teams are evolutionary

- Refactor them to avoid entropy
- Understand how your context is changing

# REA's next team horizon

- Collaboration at scale
- Evolution to platform teams





# Collaboration

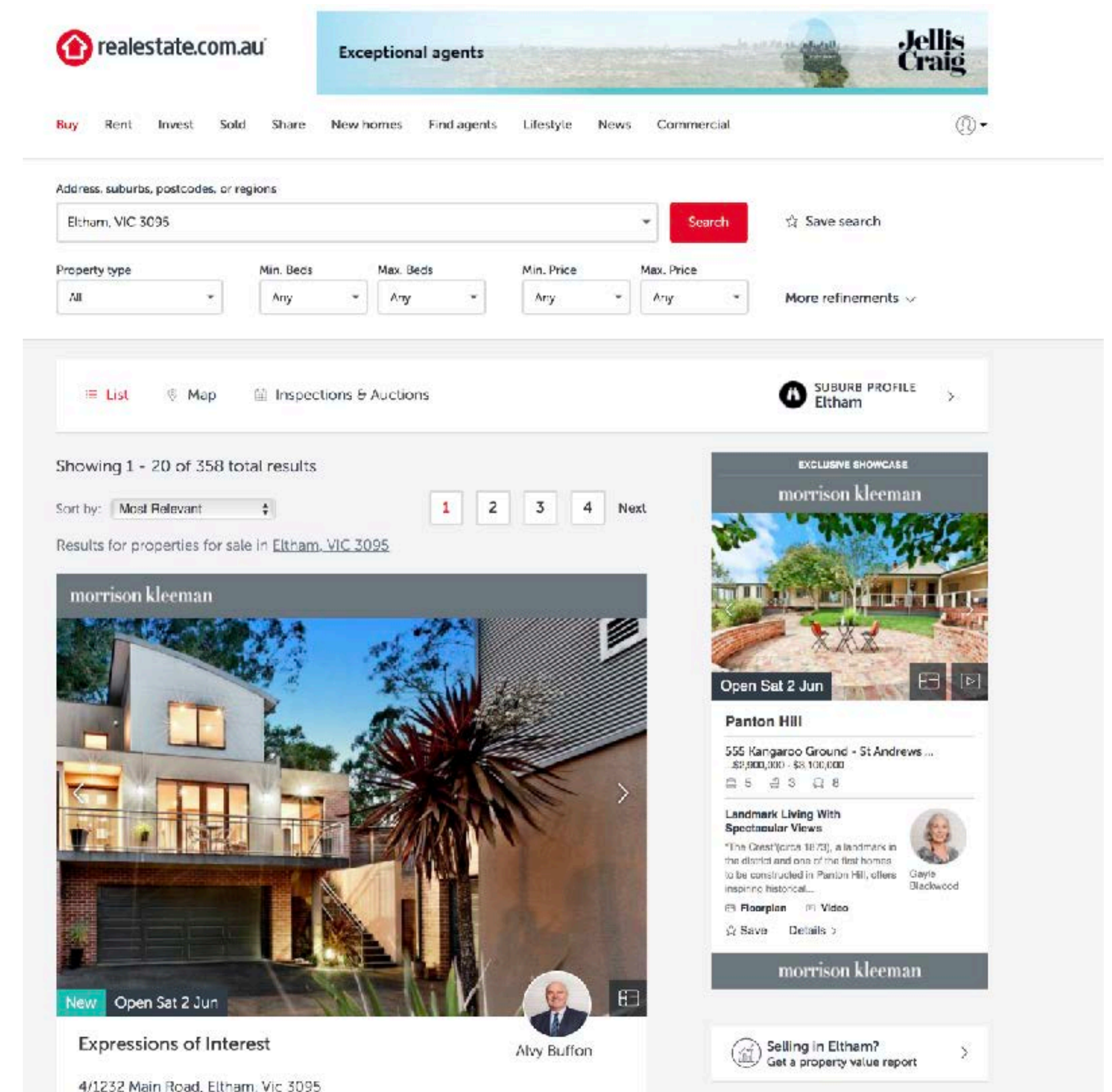
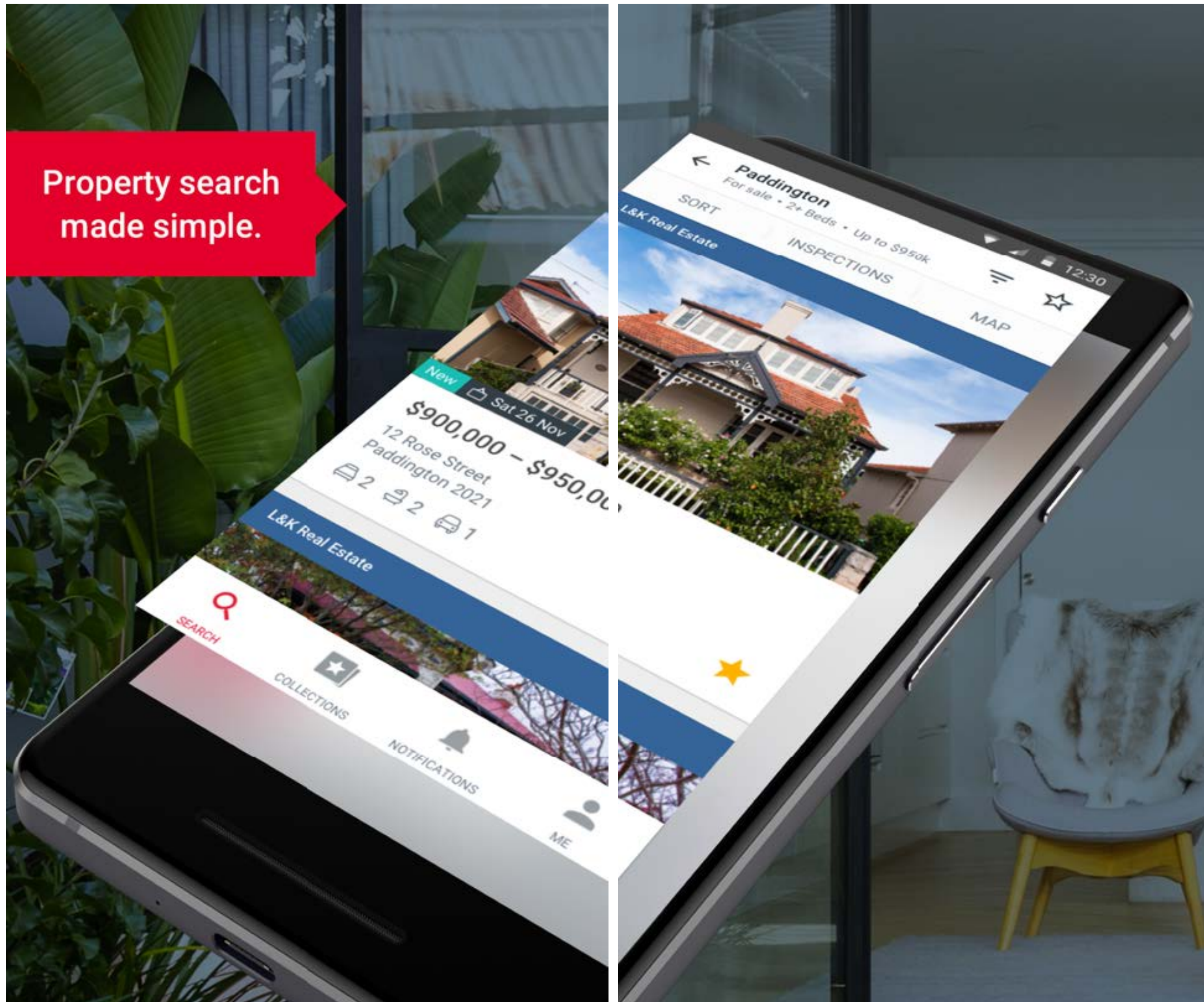
But we said teams should be autonomous. Why do we need a model for collaboration?





# A consumer platform for multiple products

Property search  
made simple.





# Enabled by federated delivery

---

- Collaboration model to scale development across REA
- Evolving to platform teams and product teams
- How will these shape our team designs?

# Key tips

- Teams are complex systems
- Teams are evolutionary. Refactor them
- Think carefully before designing teams by imitation



# Thank you

---

Peter Moran  
Engineering Manager, REA